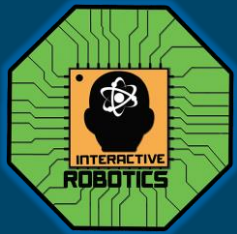


E-BOOK

IOT MQTT and Android Kodular ESP-32 DEVKIT

Created By: Iwan Muttaqin | Interactive Robotics





Iwan Muttakin

08988541995

iwancilibur@gmail.com

Community:

- Interactive Robotics
- AREI (Asosiasi Robotics Education Indonesia)

Founder at inteknus.asia

Social :

*IG/TW/FB /Github/Linkedin:
@iwancilibur*

Gaza Haikal Adzandani

089515958020

gazahaikal.2086@gmail.com

Social :

*IG/TW/FB /Github/Linkedin:
@childlimbo*



Trainer at Interactive Robotics

LINK DOWNLOAD MATERI:

<https://github.com/iwancilibur/SMA28-AGUSTUS2024>

Control dan Monitoring Sensor Aktuator dengan Blynk Platform

No	Materi	Pengetahuan	Pencapaian	Metode
1	Pengenalan Aplikasi dan Microcontroller ✓ Arduino ✓ ESP32 Devboard	Memahami macam-macam microcontroller yang dapat digunakan dalam proyek pembuatan alat-alat robotika dan IOT	Dapat menentukan microcontroller yang cocok untuk digunakan pada project yang akan dibangun	Teori
2	Mempelajari sensor khusus ✓ DHT 11 (Temperature dan Humidity)	Mengerti penggunaan dari macam-macam sensor khusus dan berikut cara merangkai dan memprogramnya.	Dapat memanfaatkan sensor-sensor jenis khusus dalam pembuatan project elektronika yang ingin dibangun	Teori dan Praktikum
3	Menghubungkan ke Aktuator dan Output ✓ Relay	Mengerti penggunaan akuator Relay sebagai penggerak objek tertentu seperti (Lampu, Motor dll.)	Dapat memanfaatkan macam-macam akuator serta dapat menganalisa fungsionalitas dari jenis akuator pada project yang akan dibangun.	Teori dan Praktikum
4	Komunikasi Jaringan ✓ Jenis-jenis Jaringan (Kabel & Wireless) ✓ Topologi Jaringan ✓ Modem, Access Point	Mengerti komunikasi jaringan dan device-device jaringan yang umum digunakan baik jaringan jenis kabel maupun wireless	Dapat memilih perangkat atau device yang digunakan untuk membangun project berdasarkan jenis jaringan yang akan digunakan.	Teori
5	Mengirim dan Menerima data dari Microcontroller ke Server MQTT Broker ✓ MQTT Publish dan Subscribe ✓ Logika Untuk Control Device Microcontroller ✓ Membangun Dashboard IOT di Android Kodular	✓ Mengerti cara mengirim data dan menerima data dari server, serta mempelajari cara membuat dashboard IOT dengan Android Kodular	Dapat memanfaatkan server sebagai penerima dan pengirim data antara device dan server serta menampilkan datanya di dashboard IOT Android Kodular	Teori dan Praktikum
6	Evaluasi			



1

Pengenalan Aplikasi dan Microcontroller

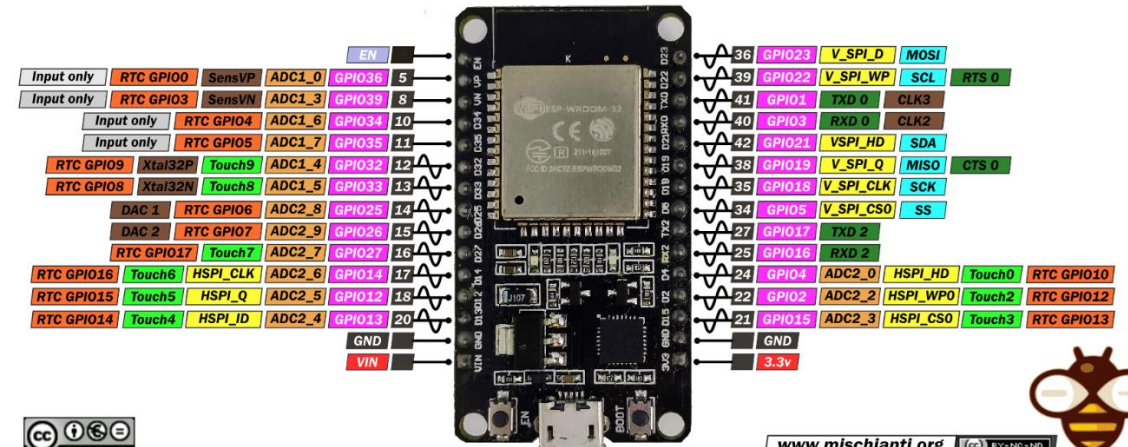
- Arduino
- ESP32 Devboard



Arduino sendiri merupakan perangkat yang bersifat **open source** dan sering digunakan untuk merancang serta membuat perangkat elektronik. Ditambah lagi **Software-nya** yang mudah digunakan dan membantu pekerjaan pengguna. Ia memiliki beberapa kegunaan, salah satunya adalah untuk mengembangkan perangkat yang dapat bekerja secara otomatis.

Mikrokontroler **ESP32** merupakan mikrokontroler SoC (System on Chip) terpadu dengan dilengkapi **WiFi** 802.11 b/g/n, **Bluetooth** versi 4.2, dan berbagai peripheral. ESP32 adalah chip yang cukup lengkap, terdapat prosesor, penyimpanan dan akses pada GPIO (General Purpose Input Output).

ESP32 DEV KIT V1 PINOUT

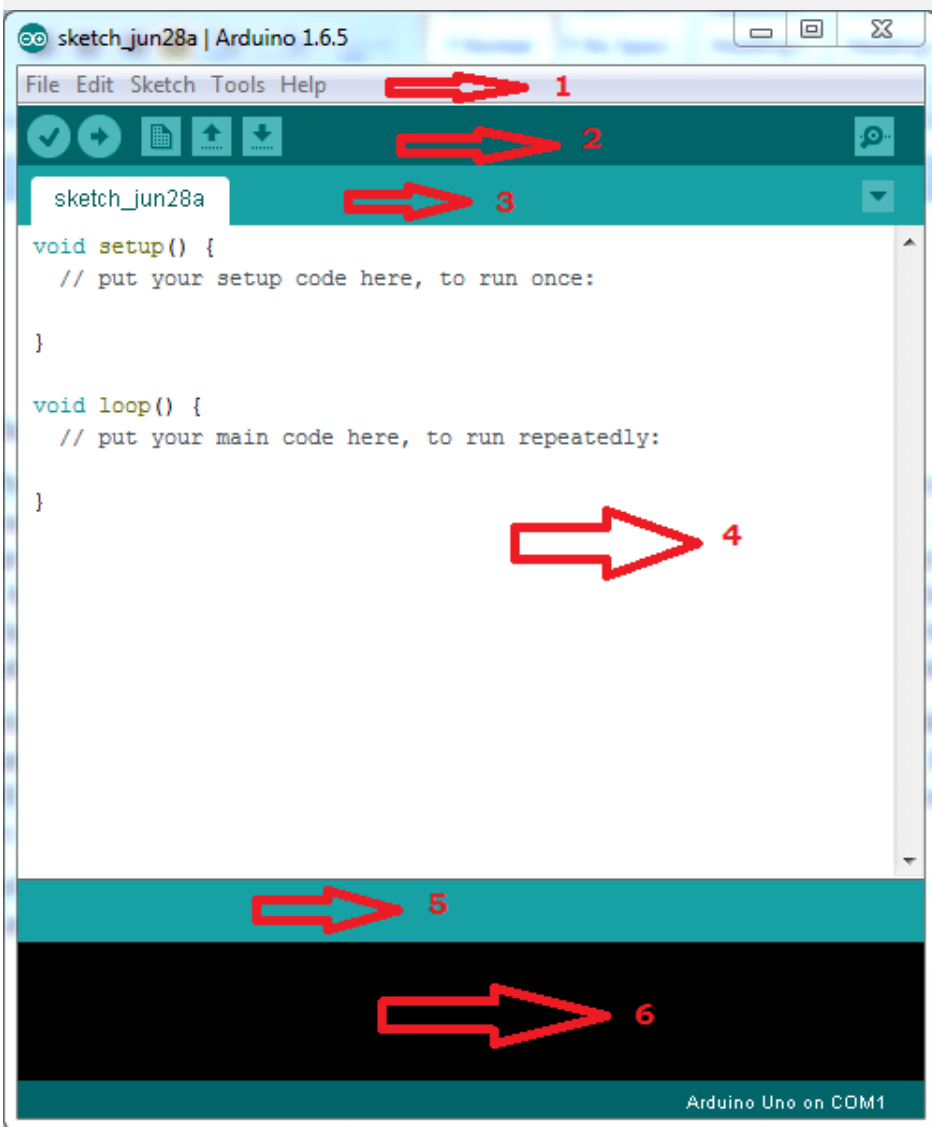


www.mischianti.org [CC BY-NC-ND]



Pengenalan Software Arduino 1

Download Software



Arduino IDE adalah software yang digunakan untuk membuat sketch pemrograman atau dengan kata lain arduino IDE sebagai media untuk pemrograman microcontroller.

Berikut adalah penjelasan bagian-bagian IDE Arduino sesuai gambar di atas :

No 1 : Menu IDE Arduino. Setidaknya ada 5 menu yang akan membantu Anda melakukan programming dengan aplikasi ini, yaitu **'File', 'Edit', 'Sketch', 'Tools' dan 'Help'**. Menu **'File'** digunakan untuk berinteraksi dengan file *.ino (**new, open, save, print, contoh program, dll**). Menu **'Edit'** digunakan untuk mengedit program yang sedang ditulis di sketch editor, Menu **'Sketch'** untuk **mem-verify, meng-upload sketch**, Menu **'Tools'** untuk memanggil tools pendukung software Arduino ini dan Menu **'Help'** berisi bantuan/ catatan/ keterangan yang mungkin Anda butuhkan jika ada pertanyaan. Tiap menu ini memiliki submenu-submenu sendiri. Silahkan dieksplor untuk lebih detail masing-masing fungsinya.

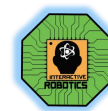
No 2 : Kumpulan shortcut yang sering digunakan dalam proses pemrograman. Dari kiri ke kanan, shortcut tersebut adalah : Verify – Upload – New Sketch, Open, Save. Dan shortcut paling kanan adalah shortcut 'Serial Monitor', yakni jendela yang digunakan untuk memonitor data yang keluar masuk di port serial Arduino (diemulasi dalam port USB ke arah komputer).

No 3 : nama sketch

No 4 : Sketch editor. Di sinilah tempat Anda menulis program/sketch Arduino dalam bahasa C.

No 5 : Jendela informasi/notifikasi jika melakukan sesuatu di IDE ini

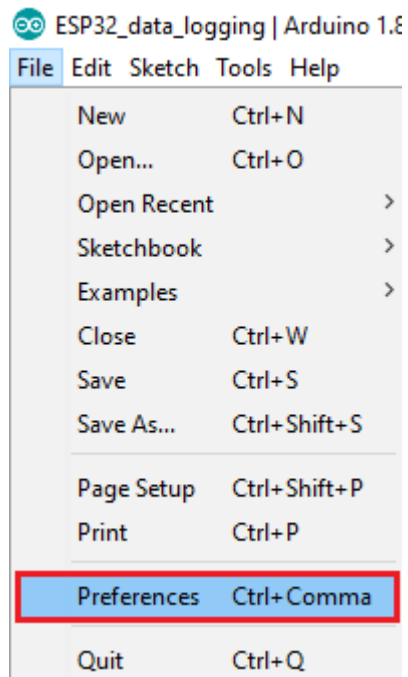
No 6 : Jendela debug. Jika ada kesalahan programming maka kesalahan tersebut akan muncul di jendela ini. Ini akan memudahkan Anda dalam mencari bug dalam program Anda



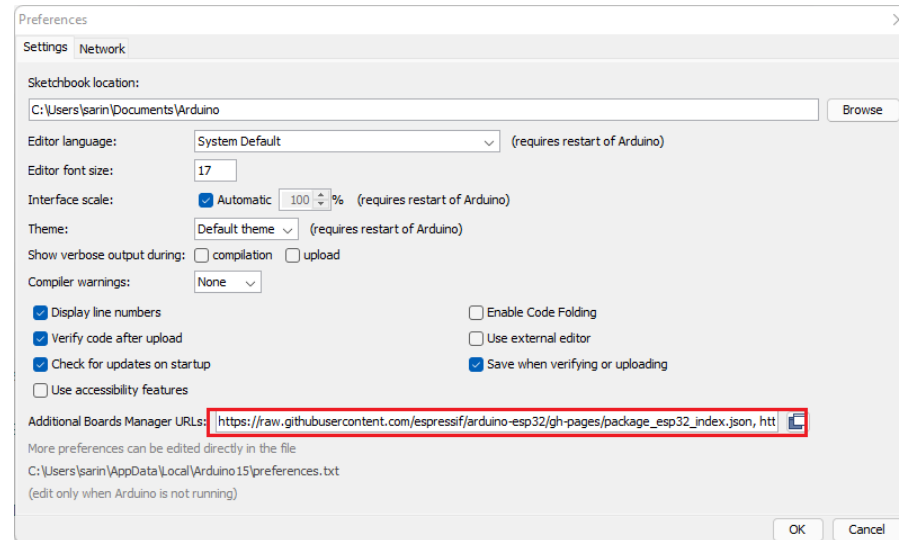
Install ESP32 on Arduino IDE

1

Langkah 1

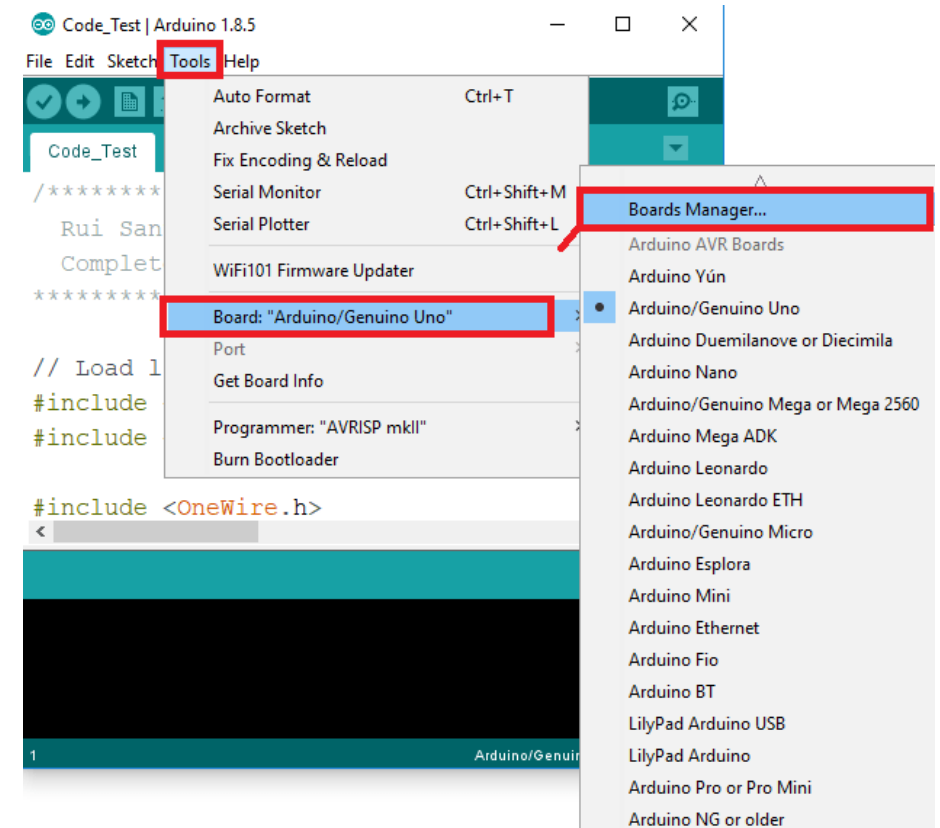


Langkah 2



Link:
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

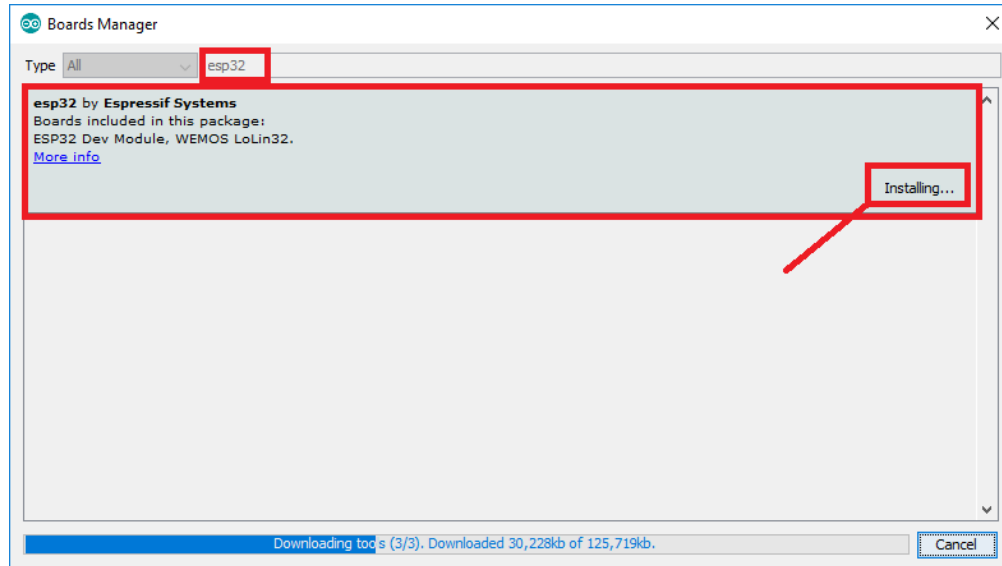
Langkah 3



Install ESP32 on Arduino IDE

1

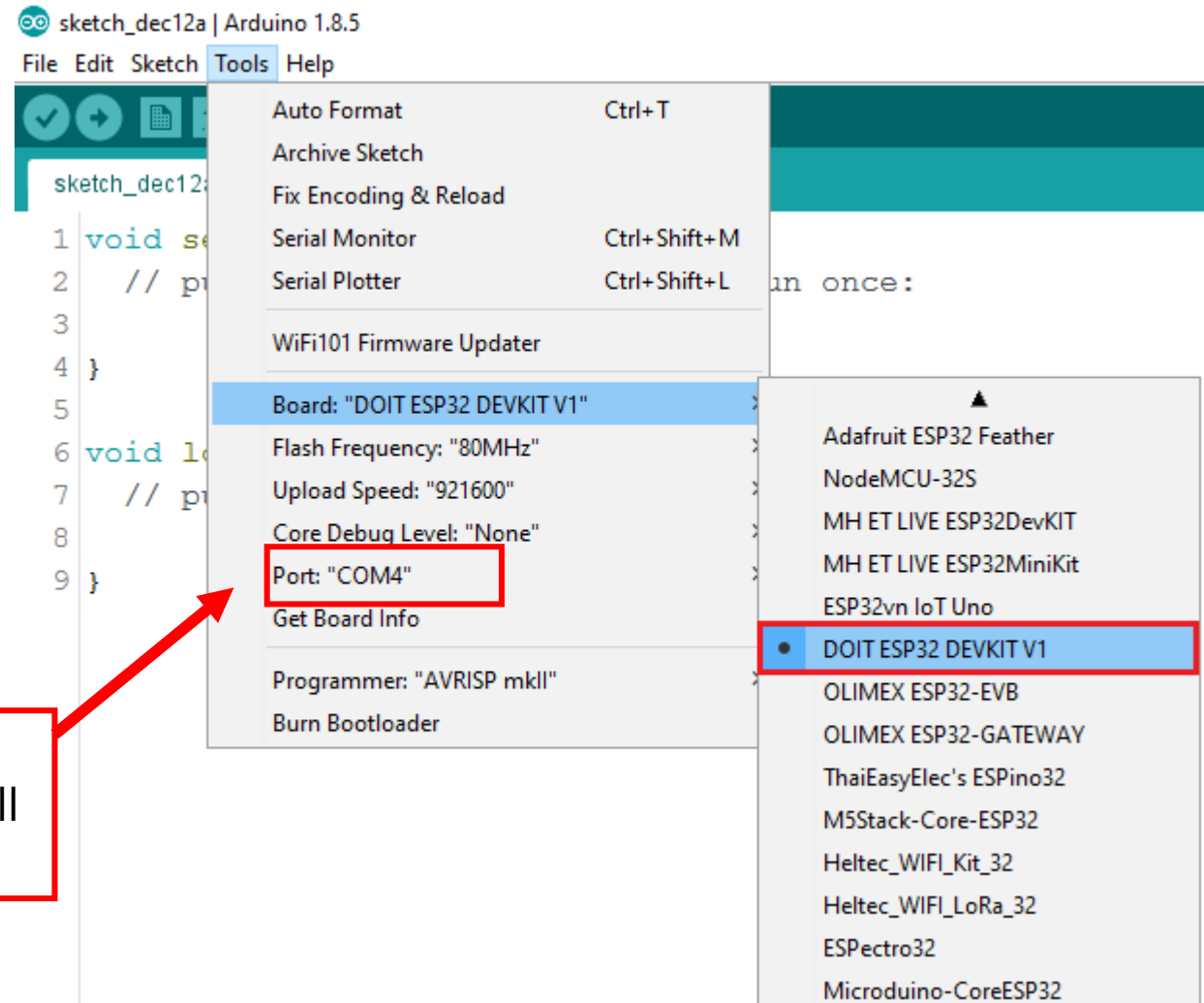
Langkah 4



Untuk **COM4** disesuaikan dengan **COM** yang terdeteksi, Apabila tidak terdeteksi silahkan install Drivernya Terlebih dahulu

Download Driver

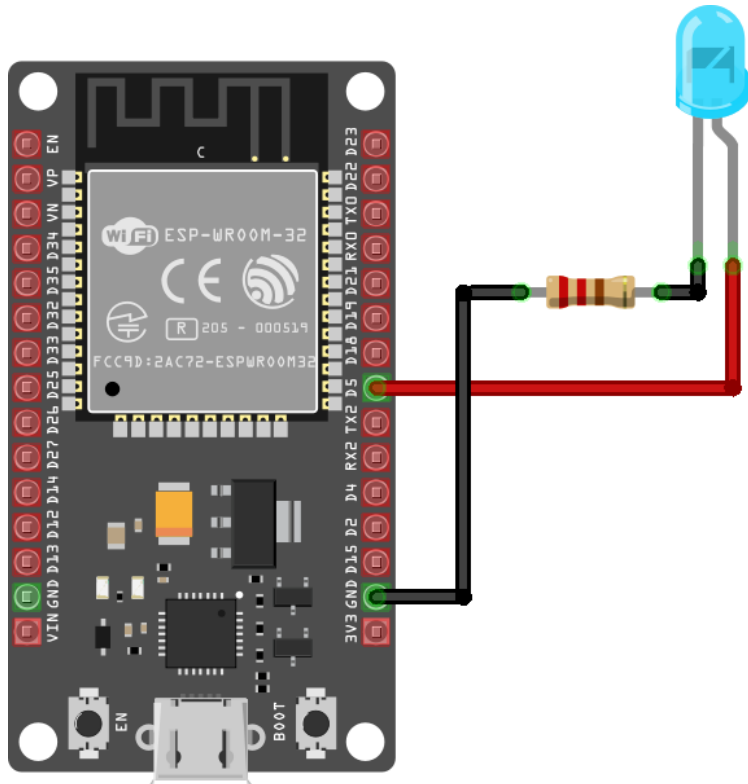
Langkah 5



"Hello Word / Blinking LED" pada ESP32 1



Apa itu Blinking LED, Membuat rangkaian LED agar LED dapat menyala selama 1 detik dan padam dalam 1 detik secara berulang, Untuk menampilkan Tulisan klik pada menu Serial Monitor.



fritzing

```
1.BLINKING_LED | Arduino 1.8.19
File Edit Sketch Tools Help

1.BLINKING_LED

byte LED = LED_BUILTIN;

void setup() {
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
}

void loop() {
  Serial.println("MENYALA ABANGKUH");
  digitalWrite(LED, HIGH);
  delay(1000);
  Serial.println("PADAM ABANGKUH");
  digitalWrite(LED, LOW);
  delay(1000);
}

Done Saving.
The sketch name had to be modified.
Sketch names must start with a letter or number,
numbers, dashes, dots and underscores. Maximum length is 63 characters.

4 DOIT ESP32 DEVKIT V1, 80MHz, 921600, None, Disabled on COM12
```



Mempelajari sensor-sensor Khusus

-DHT 11 (Temperature dan Humidity)

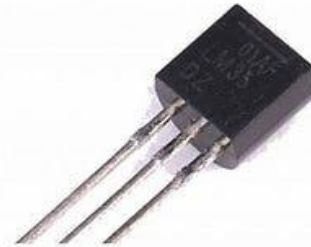
Apa yang dimaksud dengan sensor khusus, Sensor khusus merupakan sensor yang membutuhkan nilai pasti atau nilai yang sudah ditentukan satuannya, Seperti Jarak dengan satuan Centi Meter (cm), Meter (m) dan Kilo Meter (Km), atau yang lainnya seperti Suhu (Celcius) dan masih banyak lainnya, dan biasanya sensor ini sudah dibuatkan library oleh produsen pembuat sensor tersebut.



DHT11
Temp + Humidity



Thermistors



LM35



DS18B20 one-wire
Digital Temperature
Sensor



DHT22/AM2302



DS18B20

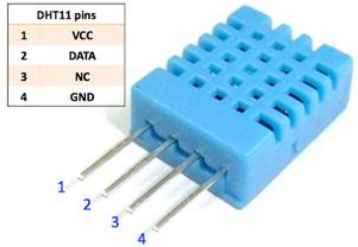
MAX6675 + K type Thermocouple
Temperature Sensor



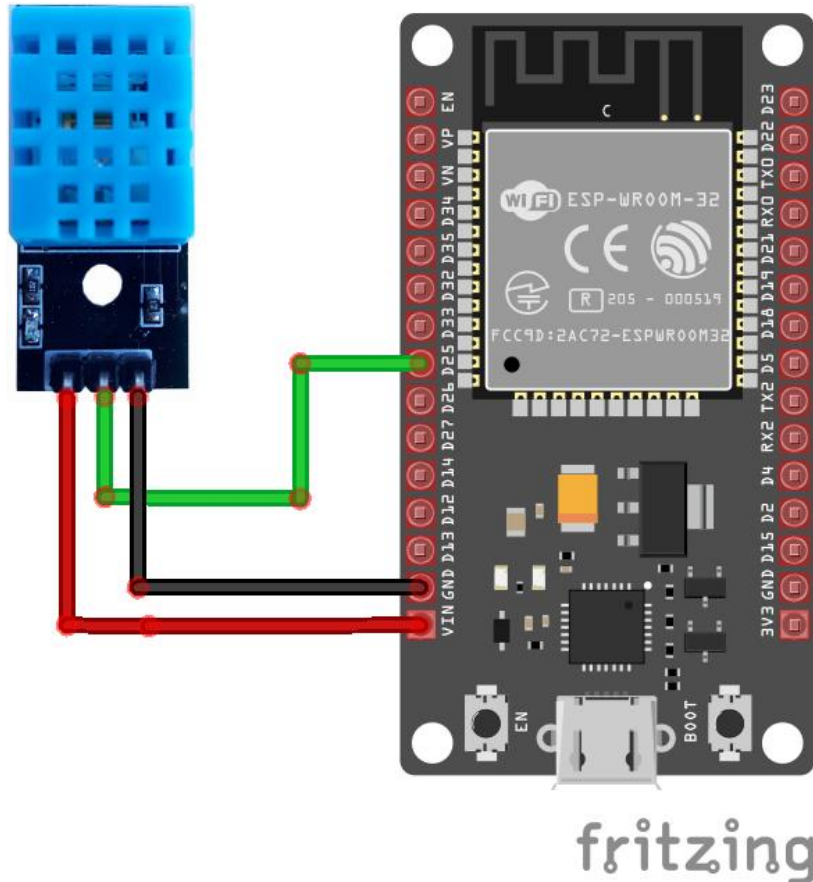
MLX90614
Infrared Temperature
Sensor

DHT 11 (Temperature dan Humidity)

2



Sensor DHT11 adalah sensor dengan kalibrasi sinyal digital yang mampu memberikan informasi suhu dan kelembaban udara. Sensor ini tergolong komponen yang memiliki tingkat stabilitas yang sangat baik serta fitur kalibrasi yang sangat akurat.



```
4_DHT11 | Arduino 1.8.19
File Edit Sketch Tools Help

4_DHT11$
#include <DHT.h>

#define DHTPIN 25
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  float suhu = dht.readTemperature();
  float kelembaban = dht.readHumidity();

  //POST DATA SERIAL
  Serial.print("SUHU      :"); Serial.print(suhu);
  Serial.print("KELEMBABAN :"); Serial.print(kelembaban);
  Serial.println();
  delay(1000);
}
```

Instal Library

- DHT
- Adafruit Unified Sensor

Chalange

Jika Suhu Lebih dari **31 C**, Maka Muncul Pesan **"Suhu Mulai Panas"**, Jika kurang dari **31 C** Muncul Pesan **"Normal"**



3

Menghubungkan ke Output Aktuator

-Relay



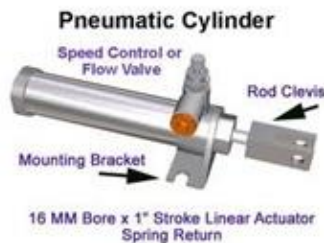
Relay



Motor DC listrik



piston



Silinder pneumatik



solenoid



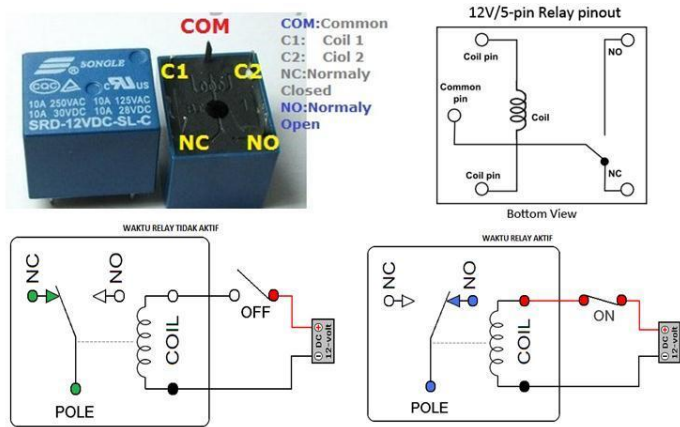
motor stepper

Aktuator adalah sebuah peralatan mekanis untuk menggerakkan atau mengontrol sebuah mekanisme atau sistem. Aktuator diaktifkan dengan menggunakan mekanisme dan elektrik biasanya digerakkan oleh signal pada tegangan 0-5 Volt, yang dikendalikan oleh media pengontrol otomatis yang terprogram di antaranya mikrokontroler.

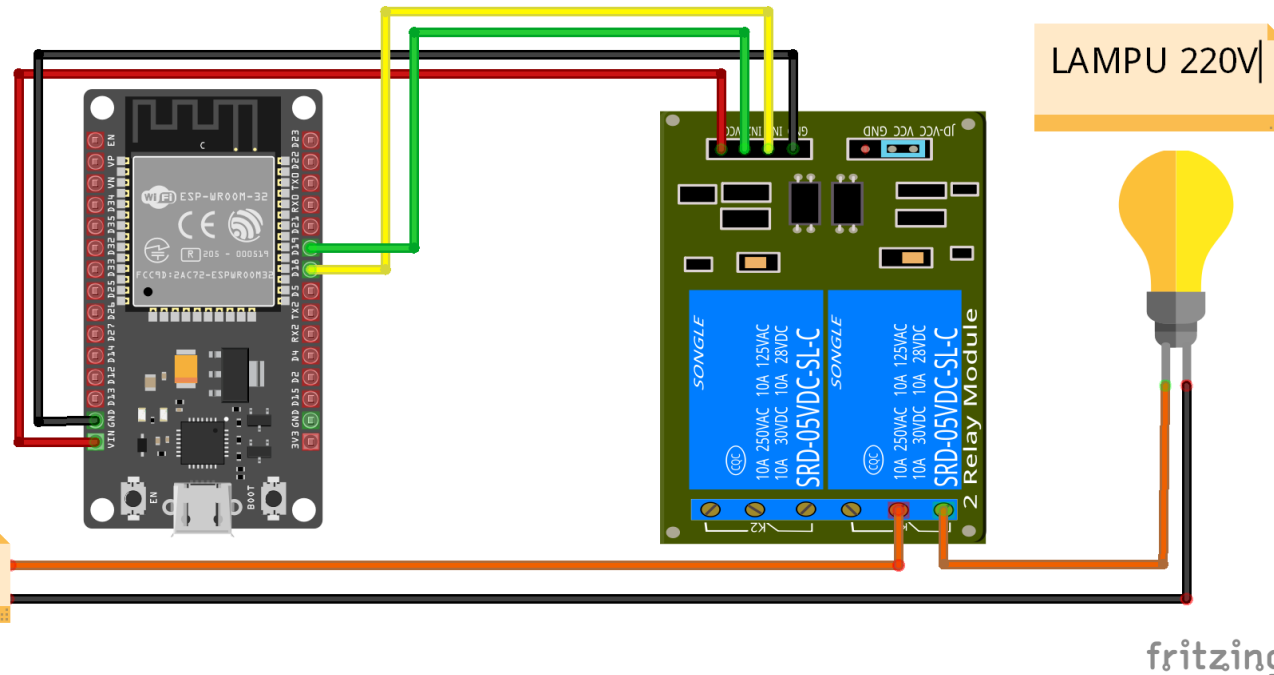
Contoh Akuator :

Relay, SSR (Solid Stat Relay), Mosfet, Motor Listrik dll.

Relay dan Lampu 220 v



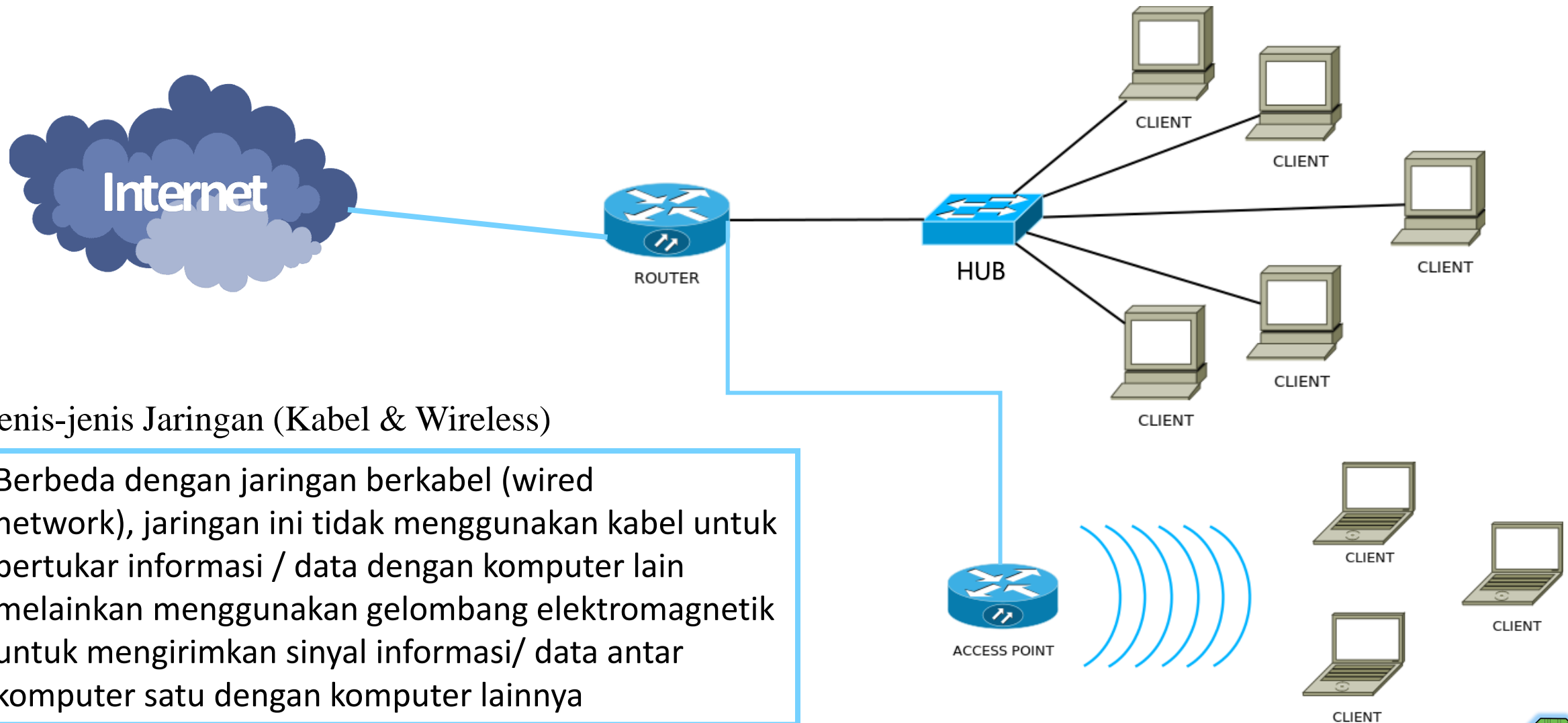
Relay biasanya digunakan untuk menggerakkan arus/tegangan yang besar (misalnya peralatan listrik 4-10 A/AC 220V) dengan memakai arus/tegangan yang kecil (misalnya 0.1 A/5 volt DC). Relay adalah komponen listrik yang bekerja berdasarkan prinsip induksi medan elektromagneti



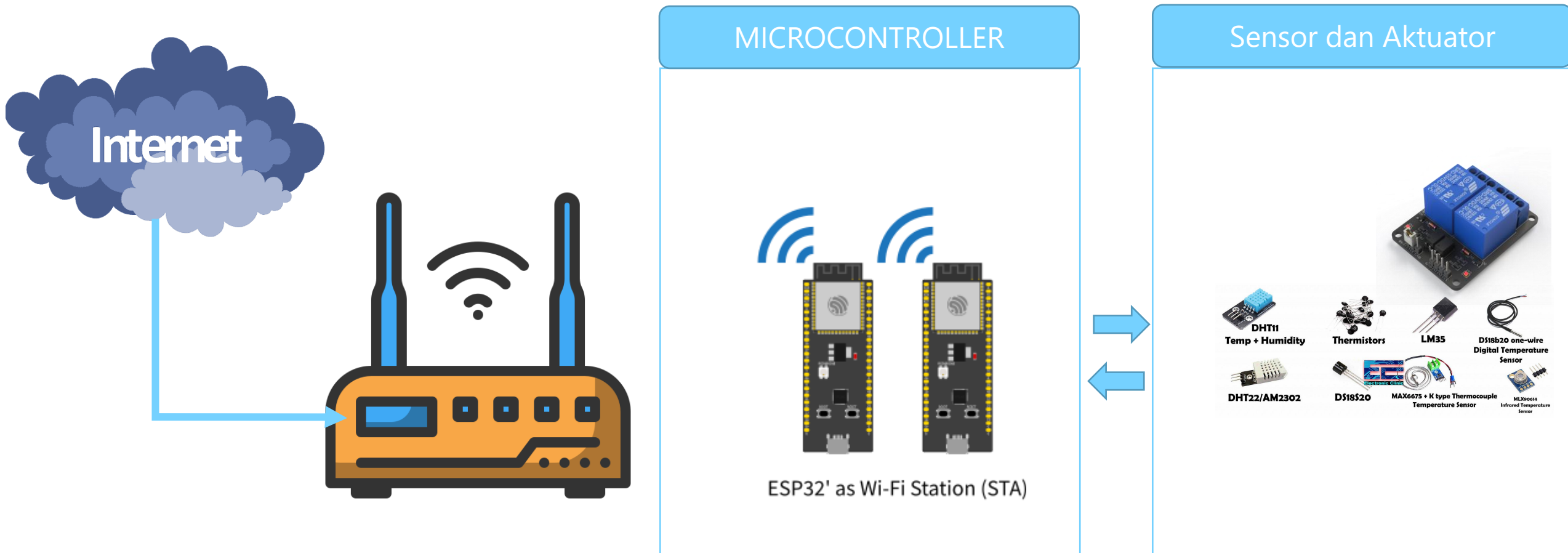
Program

Sama Seperti menggunakan LED pada materi Blinking LED

- Jenis-jenis Jaringan (Kabel & Wireless)
- Topologi Jaringan
- Modem, Access Point

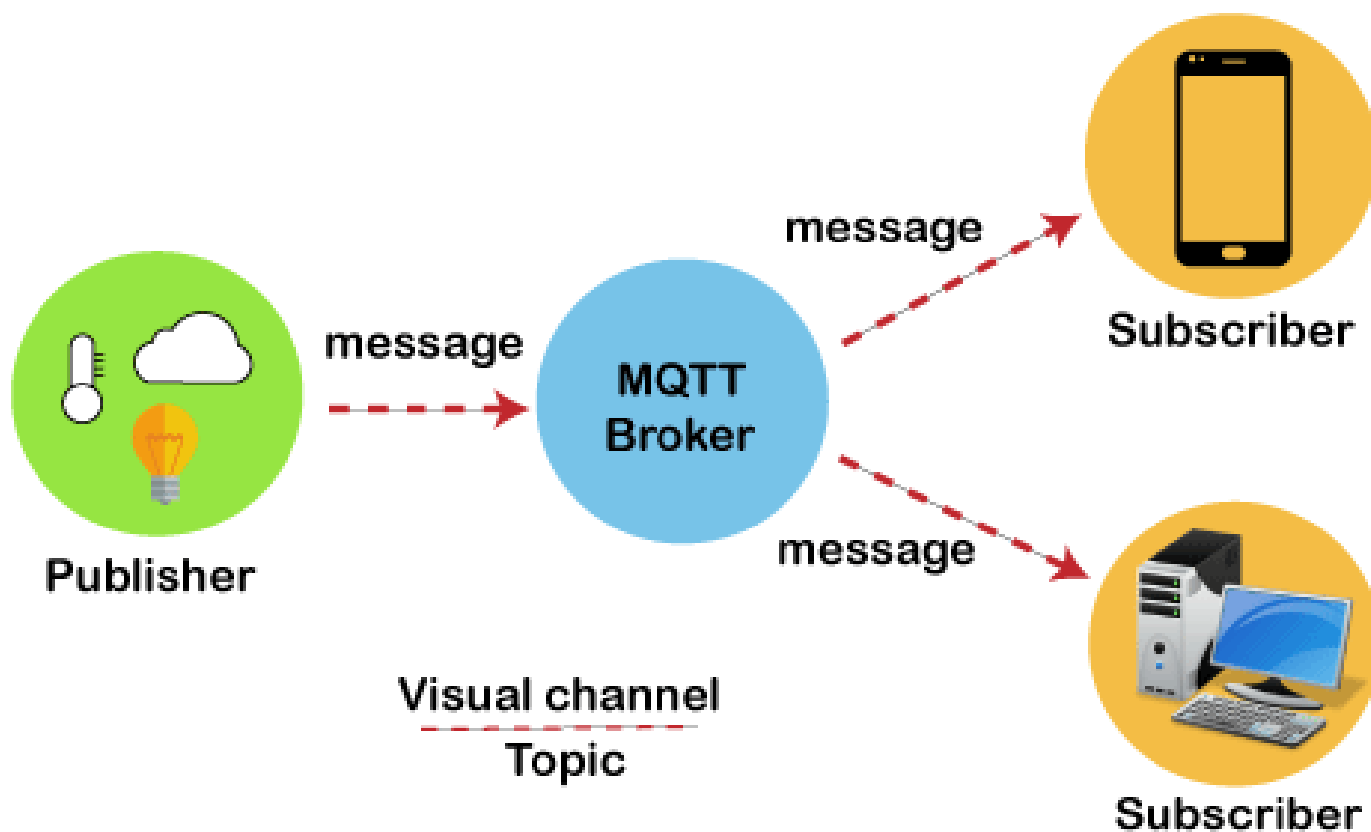


Topologi Jaringan Modem dan Acces Point



- MQTT Publish dan Subscribe
- Logika Untuk Control Device Microcontroller
- Membangun Dashboard IOT di Android Kodular
- Node RED (Dashboard)

MQTT Architecture



MQTT (Message Queuing Telemetry Transport) adalah protokol komunikasi ringan yang digunakan untuk mentransmisikan data antar perangkat, terutama dalam konteks ***Internet of Things (IoT)***. Dalam konteks MQTT, terdapat dua istilah utama yang perlu dipahami: "**Topic**" dan "**Payload**".

MQTT Publish dan Subscribe

Topic:

Definisi: Topic adalah cara untuk mengorganisir pesan-pesan dalam sistem MQTT. Ini adalah string yang digunakan untuk memberi identifikasi pada pesan atau menunjukkan ke topik tertentu.

Fungsi: Topik memungkinkan pihak-pihak yang terlibat dalam pertukaran pesan untuk menentukan area atau subjek tertentu dari data yang diinginkan atau diterima.

Contoh: Jika Anda memiliki perangkat IoT yang mengukur suhu, Anda dapat menggunakan topik seperti "***iwana/sensorsuhu/kamar1***" untuk menyampaikan data suhu dari sensor di kamar 1.

Payload:

Definisi: Payload adalah data aktual atau informasi yang dikirimkan dalam pesan MQTT setelah topik ditentukan. Ini bisa berupa teks, angka, atau format data lainnya sesuai kebutuhan aplikasi.

Fungsi: Payload berisi informasi yang ingin dikomunikasikan antar perangkat. Misalnya, dalam pesan yang membahas suhu di kamar 1, payload mungkin berisi nilai suhu aktual.

Contoh Pesan MQTT:

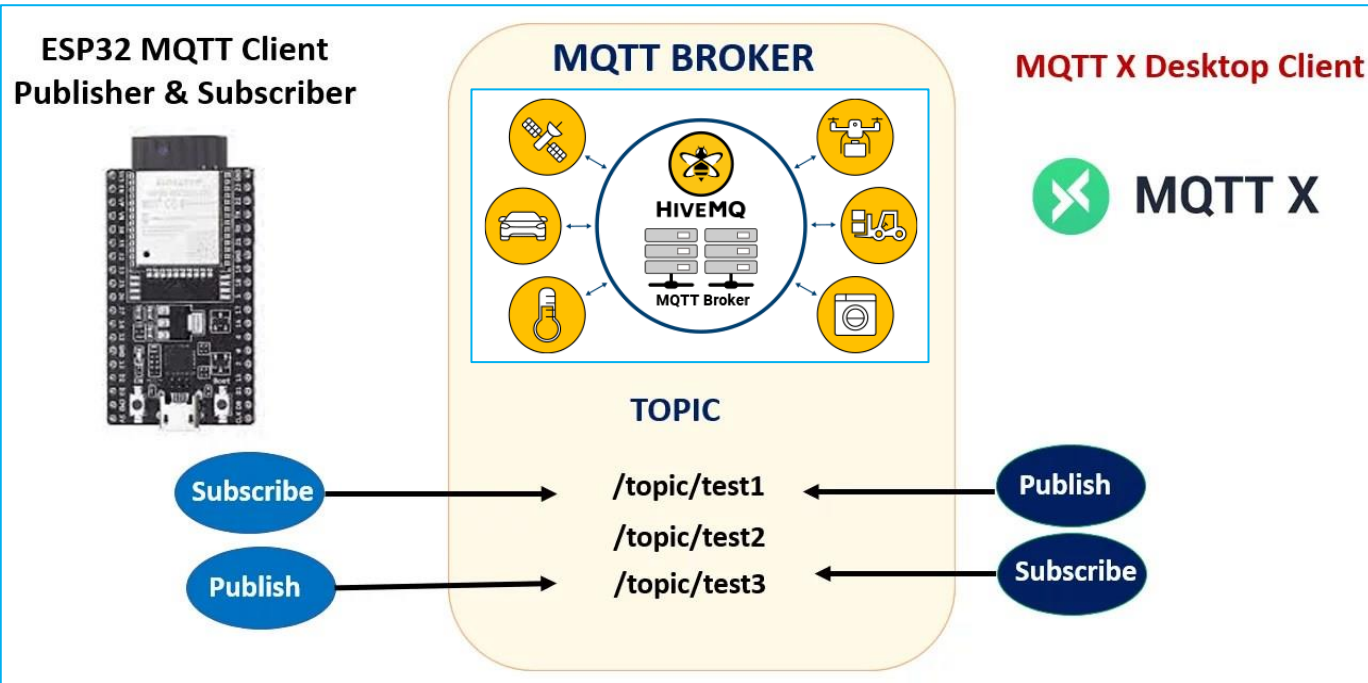
Topic : ***iwana/sensorsuhu/kamar1***

Payload : **25.5**

Dalam contoh ini, "***iwana/sensorsuhu/kamar1***" adalah **topik** yang memberi tahu penerima bahwa pesan berisi data suhu di kamar 1, sementara "**25.5**" adalah **payload** yang merupakan nilai suhu aktual yang dikirim.



MQTT Publish dan Subscribe



WEB MQTT BROKER

<https://www.hivemq.com/mqtt/public-mqtt-broker/>

Free Public MQTT Broker

Free Public MQTT Broker by HiveMQ

Our Public HiveMQ MQTT broker is open for anyone to use. Feel free to write an MQTT client that connects with this broker. We also keep a list of MQTT client libraries that can be used to connect to HiveMQ.

You can access the MQTT broker securely at:

Broker: broker.hivemq.com

TCP Port: 1883

Websocket Port: 8000

TLS TCP Port: 8883

TLS Websocket Port: 8884

[View public broker](#)

Looking for a full-featured MQTT platform?
Try HiveMQ.

Try HiveMQ as a managed service or self-hosted.

[Start free](#)

MQTT Publish dan Subscribe

Free Public MQTT Broker by HiveMQ

Our Public HiveMQ MQTT broker is open for anyone to use. Feel free to write an MQTT client that connects with this broker. We also keep a list of MQTT client libraries that can be used to connect to HiveMQ.

You can access the MQTT broker securely at:

Broker: broker.hivemq.com

TCP Port: 1883

Websocket Port: 8000

TLS TCP Port: 8883

TLS Websocket Port: 8884

[View public broker](#)

```
4_MQTT | Arduino 1.8.19
File Edit Sketch Tools Help

4_MQTT

#include <WiFi.h>
#include <MQTT.h>

const char ssid[]      = "iotgateway"; //SSID WIFI
const char pass[]      = "iotgateway"; //PASSWORD WIFI
const char* mqtt_server = "broker.hivemq.com"; //MQTT Broker
const char* client_id  = "iwancilibur"; //Harus Unix

WiFiClient net;
MQTTClient client;

unsigned long lastMillis = 0;

void connect() {
  Serial.print("checking wifi...");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(1000);
  }

  Serial.println("connecting...");
  while (!client.connect(client_id, "public", "public")) {
    Serial.print(".");
    delay(1000);
  }
}
```

Done Saving.

Invalid library found in C:\Users\iwancilibur\OneDrive - ODrive
Invalid library found in C:\Users\iwancilibur\OneDrive - ODrive
Invalid version ' ' for library in: C:\Users\iwancilibur\OneDrive

35 DOIT ESP32 DEVKIT V1, 80MHz, 921600, None, Disabled on COM12

Note:

-Install Library **MQTT.h**

-Sesuaikan WIFI SSID dan Password.

-**Client_id** Harus **Unix** tidak boleh sama dengan orang lain atau Device lain.

-**Topic** Juga harus **Unix** tidak boleh sama dengan orang lain atau Device lain.

Logika Control Device ESP32

```
Serial.println("connected!");  
client.subscribe("ping/iwancilibur");  
client.subscribe("iwancilibur/relay1");  
}  
  
void messageReceived(String &topic, String &payload) {  
    Serial.println("Topik: {" + topic + "} | Isi Pesan: {" + payload + "}");  
    if(topic == "iwancilibur/relay1" && payload=="on") {  
        Serial.println("RELAY MENYALA ABANGKUH");  
    }  
    if(topic == "iwancilibur/relay1" && payload=="off") {  
        Serial.println("RELAY PADAM ABANGKUH");  
    }  
}
```

Topic

Payload

Note:

-**Topic** untuk **Subscribe** Harus **Unix** dan tidak boleh sama dengan orang lain




Dashboard IOT di Android Kodular

Kodular merupakan aplikasi berbasis web based yang dapat membantu para pengembang dalam membangun aplikasi berbasis android. Dengan menggunakan konsep 'drag and drop', Kodular menjadi aplikasi yang sangat populer digunakan oleh pengembang dalam membangun aplikasi berbasis android.

<https://www.kodular.io>

KODULAR

Much more than a modern app creator without coding


 CREATE APPS!

+ SEE MORE

Dashboard IOT di Android Kodular

Membuat Akun Kodular

1



Sign In

Use your Kodular Account

Create Account


Sign In

Unable to login? [Reset your password](#)

Or sign in with

G

2



Sign Up

Create your Kodular Account


Already have an account?
[Sign in instead](#)

Next

Or sign up with

G

3



Almost done

Secure your account with a password

☒ I agree to the Terms of Service


<

Create account

Or sign up with

G

5



Sign In

Use your Kodular Account

Create Account

Sign In

Unable to login? [Reset your password](#)

Or sign in with

G

4

Hi inteknus,
Welcome to Kodular!

To complete your signup, please verify your account by clicking on the following link.


Confirm Account

Konfirmasi Email

Dashboard IOT di Android Kodular

Aktivasi Akun Kodular

Account verified!



Sign In

Use your Kodular Account


Enter your email*

module@inteknus.asia

Enter your password*

.....

☒ I'm not a robot



reCAPTCHA


Privacy - Terms


Create Account


Sign In


Unable to login? [Reset your password](#)

Or sign in with










Authorize Kodular Creator?

module@inteknus.asia

Kodular Creator will be able to

 View your login information

Cancel

Authorize

Legal | Junnovate, LLC

Home

KODULAR

Terms and Conditions

Terms of Service (Creator)

Privacy Policy

Cookies Policies

Disclaimer

JUNNOVATE

Corp Information

Legal Contact

innovate, LLC © 2019-2021

Kodular Creator

Last updated January 01, 2020

Terms of Use and Privacy Policy

Welcome to Kodular Creator website (the "Site"). The Site runs on Google's App Engine service. You must read and agree to these Terms of Service and Privacy Policy (collectively, the "Terms") prior to using any portion of this Site. These Terms are an agreement between you and Kodular. If you do not understand or do not agree to be bound by these Terms, please immediately exit this Site.

Kodular reserves the right to modify these Terms at any time and will publish notice of any such modifications online on this page for a reasonable period of time following such modifications, and by changing the effective date of these Terms. By continuing to access the Site after notice of such changes have been posted, you signify your agreement to be bound by them. Be sure to return to this page periodically to ensure familiarity with the most current version of these Terms.

Description of Kodular Creator

From this Site you can access Kodular Creator, which lets you develop applications for Android devices using a web browser and either a connected phone or emulator. You can also use the Site to store your work and keep track of your projects. Kodular Creator was originally developed from MIT App Inventor. The Site also includes documentation and educational content

Account Required for Use of Kodular Creator

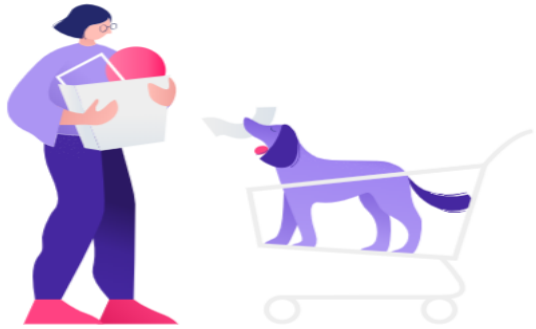
In order to log in to Kodular Creator, you need to use a Google account or provide a valid email. Your use of that account is subject to Google's Terms of Service for accounts, and the information you provide to Google is governed by Google's [Privacy Policy](#). Kodular has no access to your Google account or the information you provide for it other than the account email address, which we may use to contact you.

Information about you

I ACCEPT THE TERMS OF SERVICE



Dashboard IOT di Android Kodular



1

Create project

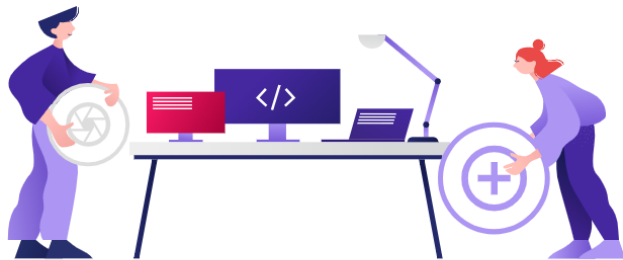
Import project

Sort by: Date modified (latest first) ▼

Welcome to Kodular!

2

Create new project



Give your new project a name

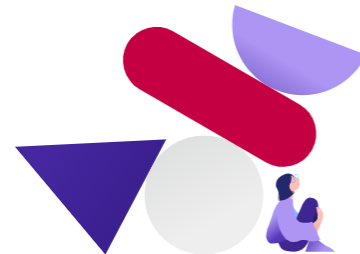
MQTT_IWAN

Cancel

Next

Configure your project

These settings can be changed later



App Name
MQTT_IWAN

Theme
Default

Minimum SDK Level
Android 5.0 - 5.0.2 (API 21)

Package Name
inteknus.asia

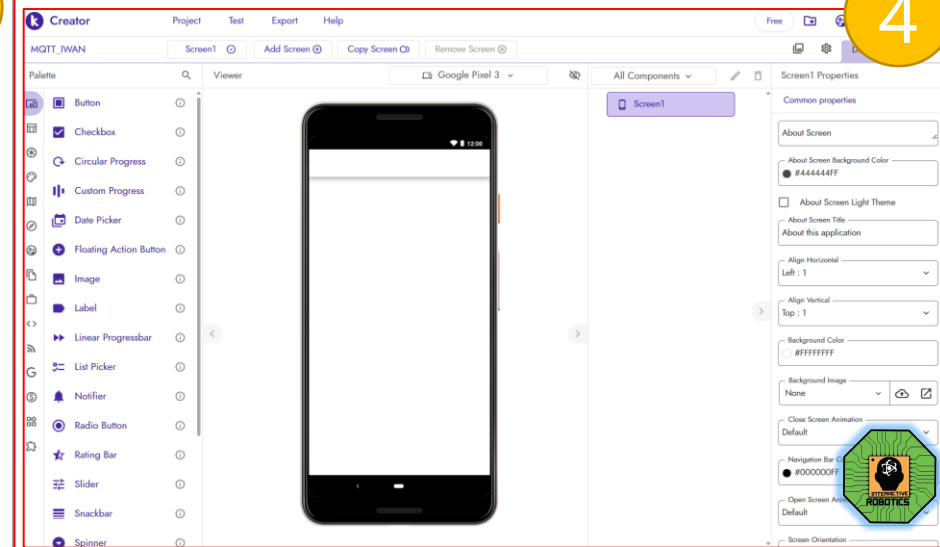
Primary Color
#3F51B5FF

Back

Cancel

Finish

3



4

Dashboard IOT di Android Kodular

Import Exstension : *UrsAI2MQTT.aix* Lokasi File ada di **Source Code**

The image illustrates the process of importing an extension into the Kodular IDE, specifically for the **UrsAI2MQTT.aix** extension. The process is divided into four numbered steps:

- Step 1:** The **Import extension** button in the left palette is highlighted with a red box and a yellow circle with the number 1.
- Step 2:** A dialog box titled **Import extension** is shown, with the **IMPORT FROM URL** button highlighted by a red box and a yellow circle with the number 2.
- Step 3:** The **UrsAI2MQTT** extension is selected in the extension palette, highlighted by a red box and a yellow circle with the number 3.
- Step 4:** The extension is added to the component palette, highlighted by a red box and a yellow circle with the number 4.

The background shows a smartphone mockup with a blank screen.

Dashboard IOT di Android Kodular

Masukan **Label**, **Button** dan Extension **UrsAi2MQTT1**
Rubahlah **Nama** dan **Text** dari masing-masing Komponen

The image displays three sequential screenshots of the Android Studio interface, illustrating the process of creating an IoT dashboard in Kodular. The interface is divided into three main sections: a preview window on the left, a component palette in the middle, and a properties panel on the right.

Step 1: The first screenshot shows the initial setup. A yellow circle with the number '1' highlights the component palette. The palette lists various components including text labels, buttons, and the 'UrsAi2MQTT1' extension. The preview window on the left shows a blank screen with a green header bar.

Step 2: The second screenshot shows the components being added to the screen. A yellow circle with the number '2' highlights the preview window. The screen now displays the text 'SUHU' and 'KELEMBAPAN', along with two buttons labeled 'RELAY 1 ON' and 'RELAY 1 OFF'. The component palette is still visible on the right.

Step 3: The third screenshot shows the properties panel for the 'Screen1' component. A yellow circle with the number '3' highlights the properties panel. The panel shows various settings for the screen, including background color, font size, font typeface, height, width, text, text alignment, text color, and visibility. The 'UrsAi2MQTT1' extension is also listed in the component palette.



Dashboard IOT di Android Kodular

Block Code

Block Code

Creator Project Test Export Help

MQTT_IWAN Screen1 Add Screen Copy Screen Remove Screen

Free

Designer Blocks

Logic Math Text Lists Dictionaries Colors Variables Procedures Screen1 text_suhu text_kelembapan btn_R1_ON btn_R1_OFF UrsAI2MQTT1 Clock1

2

when Clock1 - Timer

do

set UrsAI2MQTT1 - Broker to broker.hivemq.com

set UrsAI2MQTT1 - Port to 1883

set UrsAI2MQTT1 - Client ID to iwan/android/001

call UrsAI2MQTT1 - Connect

CleanSession true

call UrsAI2MQTT1 - Subscribe

Topic iwan/suhu

QoS 0

call UrsAI2MQTT1 - Subscribe

Topic iwan/kelembapan

QoS 0

when UrsAI2MQTT1 - PublishedReceived

do

if compare texts get Topic == iwan/suhu

then set text_suhu - Text to get Message

if compare texts get Topic == iwan/kelembapan

then set text_kelembapan - Text to get Message

when btn_R1_ON - Click

do

call UrsAI2MQTT1 - Publish

Topic iwan/relay1

Message on

when btn_R1_OFF - Click

do

call UrsAI2MQTT1 - Publish

Topic iwan/relay1

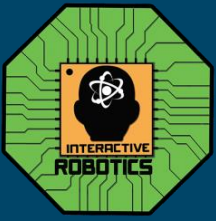
Message off

Langkah Selanjutnya
silahkan Download
Aplikasi **Companion**
Kodular di **Playstore**
Untuk mengujicoba
aplikasi yang sudah
dibuat.

Setelah selesai bisa
berkreasi untuk
menambahkan Button
dan Mempercantik
tampilan UI pada desain
Aplikasi Android.







E-BOOK IOT MQTT and Android Kodular ESP-32 DEVKIT

Created By | Interactive Robotics

A large, yellow, 3D-style speech bubble with a pointed tail pointing towards the bottom right. The words "THANK YOU" are written inside the bubble in a bold, blue, sans-serif font. The bubble has a slight shadow on the blue background.

THANK YOU